Beyond goal driven requirements engineering for purposeful system design

Colette Rolland

Beyond goal driven requirements engineering for purposeful system design

C. Rolland
Centre de Recherche en Informatique
Université Paris 1 – Panthéon Sorbonne
90, rue de Tolbiac 75013 Paris – France
E mail : rolland@univ-paris1.fr

Abstract
Requirements engineering has drawn our attention to the importance of understanding the '*whys'* of the system tot be developed. While agreeing with this position, this paper argues that the fit between the '*whats'* and the '*whys'* is key to develop purposeful systems, i.e. to ensure the best match between organisation needs (whys) and system functionality (whats). The paper explores issues related to the whys-whats relationship referred to as *the fitness relationship*.

1. Introduction

Motivation for goal-driven requirements engineering

The assumption of modern requirements engineering approaches is that understanding *Why* an information system shall be developed or reengineered is as essential as specifying *What* it shall do. The latter deals with the *system functionality* whereas the former provides its *rationale*. The argument of these approaches is that understanding the reasons why a system should provide certain functionality leads to more purposeful systems. Otherwise there is a great danger to develop systems technically sound but unable to respond to the needs of their users in an appropriate manner. Indeed, several field studies show that requirements misunderstanding are a major cause of system failure. For example, in the survey over 800 projects undertaken by 350 US companies which revealed that one third of the projects were never completed and one half succeeded only partially, poor requirements was identified as the major source of problems [Standish95]. Similarly, a recent survey over 3800 organisations in 17 European countries demonstrate that most of the perceived problems are related to requirements specification (>50%), and requirements management (50%) [ESI96].

The rationale for developing a system is to be found outside the system itself, in the enterprise [Loucopoulos94] in which the system shall function, in the *Usage World* according to the *Worlds view* [Jarke&93] shown in Figure 1. The *usage world* describes the tasks, procedures, interactions etc. performed by agents and how systems are used to do work. It can be looked upon as containing the objectives that are to be met in the organisation and which are achieved by the activities carried out by agents. The *subject world* contains knowledge of the real world domain about which the proposed system has to provide information. Requirements arise from both of these worlds. The latter imposes domain-requirements, which are facts of nature and reflect domain laws whereas the former world generates user-defined requirements, which arise from people in the organisation and reflect their goals, intentions and wishes. The *system world* is the world of system specifications in which the requirements arising from the other two worlds must be addressed. These three worlds are interrelated as shown in Figure 1. User-defined requirements are captured by the *intentional relationship*. Domain-imposed requirements are captured by the *representation relationship*.

Understanding the *intentional relationship* is essential to comprehend the reason why a system should be constructed. This relationship addresses the issue of the system purpose and relates the system to the goals and objectives of the organisation. It explains *why* the system is developed. Modelling this relationship establishes the conceptual link between the envisaged system and its changing environment. *Goal-driven approaches* have been developed to address the semiotic, social link between the usage and the system world with the hope to construct systems that meet the needs of their organisation stakeholders.

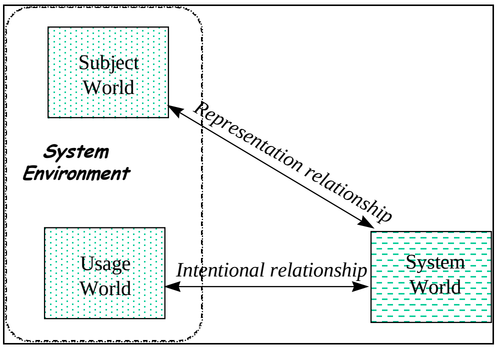

Figure 1 The relationships between the usage, subject and system worlds.

Technically, goal-driven approaches capture the *Why* in goal models taking the form of AND/OR graphs directly borrowed from problem resolution methods in Artificial Intelligence. AN/OR links are used as a refinement means. An AND-refinement link relates a goal to a set of sub-goals, which must be satisfied for the parent goal to be satisfied. An OR-refinement link relates a goal to a set of alternative refinements; this means that satisfying one of the refinements is sufficient for satisfying the parent goal. Those links are useful to establish the relationship between high-level strategic goals (i.e. transport passengers safely) and low-level operational goals (i.e. keep doors closed when moving) expressing requirements on the system. They also help *explore alternative choices* [Rolland&99; Lamsweerde00; Yu94), *detect conflicts* [Lamsweerde00a; Robinson89] and ensure *requirements pre-traceability* [Gotel&94; Pohl96; Ramesh95].

Towards the fitness relationship: a direct relationship between goals and system functionality

Despite establishing a relationship between high-level enterprise goals and system requirements, AND/OR graphs don't permit to relate easily those goals to the system functionality. First, AND/OR goal graphs could consist of hundreds of goals and secondly, each system function is across several requirements. Thus, the link between high-level strategic goals, which reflect the organisation mission and objectives with the system functionality, is far to be a direct one. This does not facilitate solving the alignment problem between organisational goals and system functionality, particularly in situations of change because the propagation of organisational changes into the system functionality needs to explore a large set of complex links.

Our position is that to solve the alignment problem between organisational goals and system functionality in the current prevalent context of rapid change a *direct, strict defined coupling* between the latter and the former is required. Such a *direct relationship, which we refer to as the fitness relationship*, implies a modelling formalism that should provide two faces, one for understanding the system functionality viewpoint and the other for the business viewpoint. This should avoid the '*conceptual mismatch'* [Arsajani01] identified as a major source of difficulty in attaining the alignment in practice [Eatock00], [Reich96]. We experienced this issue in different projects [Rolland&03] including ERP installation projects [Rolland&00; Rolland&01] where we found that ERP experts and organisational stakeholders had difficulty to match each other requirements. Indeed, the customising process typically focuses on the ERP functionality and on its customisation. The functionality is expressed in low-level details such as data to be maintained and operations to be carried out whereas organisations think in terms of their goals and objectives. This results in a language mismatch between ERP experts and organisation stakeholders. This mismatch exposes the ERP system installation to the danger of failing to meet the requirements of organisations.

In this paper we introduce the *Map* representation formalism and discuss its potential to (a) *model the fitness relationship*, (b) *refine* it, i.e. express it at different levels of detail, (c) *explore alignment alternatives* and (d) *support a customising process.*

In the rest of this paper we introduce first the *Map* and then, consider each of the four issues.

2. Introducing the Map

In this section we introduce the concept of a map and illustrate it with the *Energy* product. *Energy* is a fictitious COTS product, however inspired by a real IT product to support electricity distribution processes.

The *Map* representation system allows representing a process model expressed in intentional terms. It provides a representation mechanism based on a non-deterministic ordering of *intentions* and *strategies*. We will use it as a means for representing *Energy* goals/strategies and as a basis for aligning *Energy* functions to organisational requirements.

A *map* is a labelled directed graph (Fig.2) with *intentions* as nodes and *strategies* as edges between intentions. An edge enters a node if its strategy can be used to achieve the intention of the node. Since there can be multiple edges entering a node, the map is capable of representing many strategies that can be used for achieving an intention.

An *intention* is a goal that can be achieved by the performance of a process. For example, the *Energy* map in Fig. 2 has *Serve Customer Request* and *Sell Electricity* as intentions. Furthermore, each map has two special intentions, *Start* and *Stop*, to respectively start and end the process.

A *strategy* is an approach, a manner to achieve an intention. In Fig.2 *Competitive strategy* is a manner to *Serve Customer Request* in a context of deregulated energy distribution.

A *section* is a key element of a map. It is a triplet as for instance <*Start, Serve Customer Request, Competitive strategy*> which couples a source intention (*Start*) to a target intention (*Serve Customer Request*) through a strategy (*Competitive strategy*) and represents a way to achieve the target intention *Serve Customer Request* from the source intention *Start* following the *Competitive Strategy*. Each section of the map captures the condition to achieve a business goal and the specific manner in which the process associated with this goal can be performed.

Sections of a map are *connected* to one another. This occurs:

(a) *When a given intention can be achieved using different strategies*. This is represented in the map by several sections between a pair of intentions. Such a map topology is called a *multi-thread*. In Fig.2, the two strategies to achieve the *Serve Customer Request* intention namely *Competitive strategy* and *Captive strategy* constitute a *multi-thread*. These are two alternative ways to serve 'eligible' and 'non eligible' customers, respectively. Eligible customers have the choice of their energy distributor whereas 'non eligible' customers remain captive of the company.

(b) *When an intention can be achieved by several combinations of strategies*. This is represented in the map by a pair of intentions connected by several sequences of sections. Such a topology is called a *multi-path*. In general, a map is a multi-path from *Start* to *Stop* and contains *multi-threads*. Fig.2 contains two paths from *Start* to *Stop* to handle (a) the traditional management of electricity provision through the *Serve Customer Request by Captive strategy and* the *Sell Electricity by Credit strategy* intentions & strategies and (b) the electricity provision management in a deregulated context through the *Serve Customer Request by Competitive strategy and* the *Sell Electricity by Advance Payment strategy* intentions & strategies.

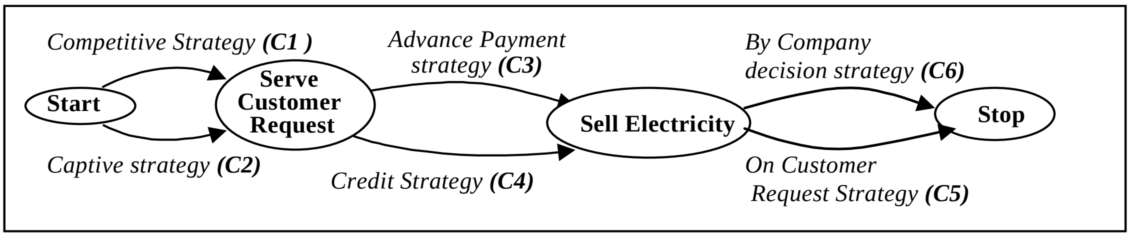

Figure 2 The Energy Map

As a consequence of the multi-path and multi-thread topologies, a map does not represent a flow of tasks but a non-deterministic ordering of goal/strategy selections.

3. Modelling the fitness relationship

In this section we use the *Energy* map to show how the map allows a uniform representation of business goals and system functionality, thus providing a means to model the *fitness relationship*. The coupling between business goals and system functionality is achieved by *simply relating each section of a map to a system component*. Therefore any section can be regarded from two viewpoints: the *business viewpoint* and the *system viewpoint*.

For example, the *Energy* map in Fig.2 contains 6 sections, C1 to C6. Every section of the map represents both (a) a function provided by the *Energy* COTS product and (b) the business goal that can be satisfied using this function.

Let us consider the multi-thread between the *Serve Customer Request* and *Sell Electricity* intentions in the *Energy* map. There are two different strategies namely the *Advance Payment strategy*, and the *Credit strategy*, to achieve the organisational goal *Sell Electricity*. On one hand, from the *business viewpoint*, these strategies identify two rather different business ways to get the customer to pay for his electricity consumption. Indeed the *Advance Payment strategy* refers to a solution based on the use of payment cards to energise the customer meter whereas the *Credit strategy* refers to the more conventional solution where the electricity company provides electricity to its customer and gets paid after consumption. On the other hand, from the *system viewpoint*, these two strategies identify two components of the COTS product that are in this case, variations of the same function. Each variation is simply captured in the map section. The triplet <*Serve Customer Request*, *Sell Electricity*, *Credit strategy*> is an example of section in the *Energy* map which corresponds to one COTS component to "*Sell Electricity*".

More generally, every section in a map is associated with a component. Therefore, from the *system viewpoint* a map can be seen as a means to explain how a complex component is made of other components and in which way components co-operate to achieve collectively an organisational goal. From the *business viewpoint*, a map highlights the business goals together with their associated strategies, which can be supported by system components.

Table 1 briefly describes the six components of the *Energy* COTS product associated with the six sections of the map as indicated (with component's reference) in Figure 2.

Ref	Component Name	Component Interface	Component Body
C1	Customer servicing in a competitive environment	<(As-Is model), Serve Customer Request with Competitive strategy>	Provides IT support to -install an IFD to serve customer request -develop customer culture within the company and measure customer satisfaction -contract customers -market the company
C2	Customer servicing in a captive environment	<(As-Is model), Serve Customer Request with Captive strategy>	Provides IT support to -handle customer requests -keep track of customer complaints -manage customers and customer installations
C3	Electricity selling in an open market	<(Customer Id), Sell Electricity with Advance Payment strategy>	Provides IT support to install card based meters and keep track of their use
C4	Electricity selling in a conventional way	<(Customer meter Id), Sell Electricity with Credit strategy>	Provides IT support to manage the process chain of conventional meter reading, electricity consumption billing and payment collection (see Figure 3)
C5	Stopping electricity provision on customer request	<(Customer Id), Stop on Customer Request strategy>	Provides IT support for customer disconnection on customer request
C6	Stopping electricity provision on company decision	<(Customer Id), Stop by Company Decision strategy>	Provides IT support for customer disconnection by company decision

Table 1 List of Energy components

The description of the six components in Table 1 shows that each component has an *interface* and a *body*. The body is what the component really does whereas the interface is the visible part of the component. The body of the component C1 for example, provides IT support to install an intelligent front desk to deal with the various customer requests. Its interface is a couple <situation, intention> stating the precondition for the component to be used (the situation is that the current state of the company has been modelled in a so-called As-Is model) and the intention that can be fulfilled in that situation (to *Serve Customer Request*) following a given strategy, namely the *Competitive strategy*.

Thus, there is a tight connection between the components and the map :

(a) Each section in the map is associated to a component,
(b) The interface intention of the component is the target intention of the section completed by the name of the section strategy and,

(c)The interface situation refers to the state resulting from the fulfilment of the section source intention.

As shown above with the *Energy* example, the Map representation formalism provides a means to model the *fitness relationship:*

- At a fine grain through the correspondence between a section in a map and a system component,
- At the gross grain of the entire map by associating the business process as a whole to an assembly of system components.

4. Refining the fitness relationship

Refinement is an abstraction mechanism by which a given entity is viewed as a set of interrelated entities. Refinement is known as a means to handle complexity. Our belief is that such *refinement mechanism* is required for *handling the fitness relationship* in *a systematic, controlled manner.* Indeed, it would be inconvenient to view in one shot, a fitness relationship as one monolithic, flat structure. A layered approach may help mastering progressively the complexity of the relationship. This confirms our experiences, which show that the refinement ratio is around 20, meaning that a relationship, initially seen as a whole, finally leads to a complex organization of about 20 sub-relationships.

From our knowledge, there are very few attempts to provide a refinement mechanism of the fitness relationship [Potts97]. As mentioned in the introduction, goal driven approaches capture the objectives of a system at various levels of abstraction in goal graphs and goal decomposition is used to relate high-level goals to low level, operationalisable goals. These are leaves of the goal graph that point out to the requirements for a given functionality. One can therefore see that goal decomposition does not support a top-down reasoning about the fitness relationship. Instead goal decomposition is a mechanism leading to the establishment of the fitness relationship as the link between system functionality and several leaves of the goal graph.

In the map approach we defined a refinement mechanism in order to *refine a section* of a map at level *i* into an *entire map* at a lower level *i+1*. Therefore, a fitness relationship (captured in a section of the map) is refined as a complex graph of sections, each of them corresponding to sub-relationships between the business and the system. Therefore, what is refined by the refinement mechanism offered by maps is in fact the fitness relationship itself. We found this mechanism helpful to understanding the fitness relationship at different levels of detail.

Let us exemplify this mechanism by refining the section C4, *Sell electricity with Credit strategy* of the *Energy* map presented in Fig. 2. The C4 refined map M4 is shown in Fig. 3.

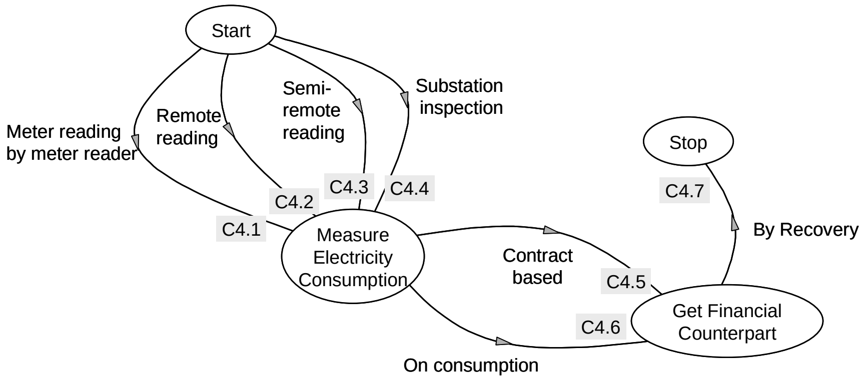

Figure 3 M4: Refined map to Sell Electricity by Credit strategy

From the *business viewpoint*, the refined map M4 explains how the *Sell Electricity by Credit strategy* intention is refined by a graph of goals and associated strategies. It identifies two key intentions, namely *Measure Electricity Consumption* and *Get Financial Counterpart* and indicates an ordering constraint: in order to bill the customer it is necessary to measure first his/her consumption. The map also shows that there are several ways to achieve each of these two intentions. For example, there are four strategies to *Measure Electricity Consumption* ranging from the classical *Meter Reading by Meter Readers* to a completely innovative way based on *Remote Meter Reading*. The former is based on manual readings taken by meter readers visiting customer sites whereas the latter is automatically done but requires special meters to be installed. Similarly, there are two ways for *Getting Financial Counterpart*, by *Contract Based strategy* or *Consumption Based strategy*.

From the system viewpoint, this map explains how the complex component C4 is made of other components and in which way those co-operate to achieve collectively the C4 business goal. Fig. 3 shows that there are 7 sub-components, C4.1 to C4.7 of the C4 component that shall cooperate as indicated by the multi-thread and multi-path topology of the C4 refined map.

Since refinement results in a map, it produces a multi-thread, multi-path structure at level *i+1*. As a result, for a given section at level *i*, not only (1) multiple threads describe *alternative sub-sections* at level *i+1*, but also (2) the multi-path structure introduces several different *combinations of sub-sections*. Therefore, section refinement is a more complex structure than a simple composition structure such as AND/OR goal decomposition Indeed, it provides at the same time (a) several alternative decompositions of the initial fitness relationship into its constituents, and (b) different alternatives to its constituents themselves. We found this mechanism useful in practice as a means to reason on the alignment at different level of detail and to fine tune the selection of the adequate system sub-functions in an ERP customizing process.

5. Exploring alternative fitness relationships

When proceeding to the alignment a business goal and a system functionality, it is necessary to explore of the different ways to establish the link; in other words to investigate the *alternative fitness relationships* for the problem at hand. In a change perspective this is crucial for the envisionment of the future system.

Most of the goal based requirements engineering approaches recognize the usefulness of goals in exploring alternative designs [Anton98], [Yu01], [Rolland&98], [Paech02]. This is generally achieved using AND/OR refinement where “alternative goal refinements [expressed with OR links] allow alternative system proposals to be explored” [Letier02]. However, if

alternative goals help reasoning about alternative system functionalities to achieve the parent goal, the issue of exploring fitness relationship alternatives raises the question of reasoning about alternative combinations of functionalities across the entire AND/OR goal graph.
We found that maps, as a means for describing alternative complex assemblies of functionalities, can help in this exploration and in the discovery of the ones that best fit the business goals. The multi-thread topology of maps corresponds to OR structures in a goal graph. In addition, the multi-path map topology helps reasoning and evaluating alternative assemblies of functionalities. Such assemblies give rise to a *payoff analysis*. The result is the selection of sections that show the combination of the required functions.

For example, the map shown in Fig. 2 identifies seven different functions for the management of electricity supply in a utility company. Each function is identified by a section in the map. The <*Start, Sell electricity, with credit strategy*> section identifies C4 for selling electricity in a conventional way, which provides IT support to manage the process chain of conventional meter reading, electricity consumption billing and payment collection with seven sub-component as the refined map M4 in Fig.3 shows it.

Once C4 has been selected, one has to decide on how electricity should be measured and how the financial counterpart should be obtained. Each sub-component (component in M4) selection has however a payoff that can be analysed in the view of its combination to another component. The pay-off analysis for C4 sub-components is summarized in the Table2 below.

<table>
<tr><th></th><th></th><th colspan="2">5.1. Get Financial Counterpart</th></tr>
<tr><th></th><th></th><th>Contract based</th><th>On consumption</th></tr>
<tr><td rowspan="4">Measure electricity consumption</td><td>Meter reading by meter reader</td><td>Can be envisaged at sustainable cost if visits are achieved at a low frequency (e.g. once or twice a year)</td><td>Excluded because too difficult to organise all visits at the required pace.</td></tr>
<tr><td>Remote reading</td><td colspan="2">Cost effective combination that can be done in real time. However, remote reading is not completely secure. A complimentary check of electricity measurement is thus needed, e.g. by meter reader, or by substation inspection.</td></tr>
<tr><td>Semi-remote reading</td><td>Cost effectiveness is a linear function of the number of contracts per cluster of semi-remote reader.</td><td>Very costly if the number of customers paying on consumption, per cluster of remote reader is low</td></tr>
<tr><td>Substation inspection</td><td>Only possible if the connected meter readers relate to single contract. Otherwise, calls for individual reading.</td><td>Cost effective way to handle the verification of consumers invoiced by remote reading clustered on the same substation.</td></tr>
</table>

Table 2 Pay off summary for the selection of C4 sub components

Let us consider the case where it is necessary to *Get Financial Counterparts* both contract based and on consumption. The table shows that remote readings are a cost effective way to handle electricity measurement in both cases. Indeed, it is real time and therefore adapted to payment on consumption. Besides, the cost of installing remote readers can be included in the contract prices and recovered in the long term. However, the payoff table also says that remote reading, as it is automated, is not fully reliable and should be double-checked, e.g. by using subsection inspection. New opportunities emerge from further analysis. For example, remote reading makes it possible to analyse consumption in real time; new types of contracts can also be proposed to automatically adapt electricity production to consumption. This would be useful when electricity provision should be reliable, e.g. for refrigeration warehouses, hospitals, telephony service, etc.

6. Supporting the customising process

In this section we address partially the problem of aligning a COTS product functionality to the organisational needs and illustrate briefly with the installation of *Energy* COTS product in a company which is still working under a monopolistic mode of distributing energy to its customers. We view the customising process as a change handling process that creates a movement from an existing situation, captured in the *As-Is model*, to a new one reflecting the set of user requirements for the future, captured in the *To-Be model*. We propose to express these models as maps. This is reflected in Fig. 4 by (a) *Construct As-Is Map* and (b) *Construct To-Be Map*. Furthermore, we extend this position to cope with the specificity of ERP installation by introducing (c) *Construct ERP functionality Map* and (d) *Matched Map*.

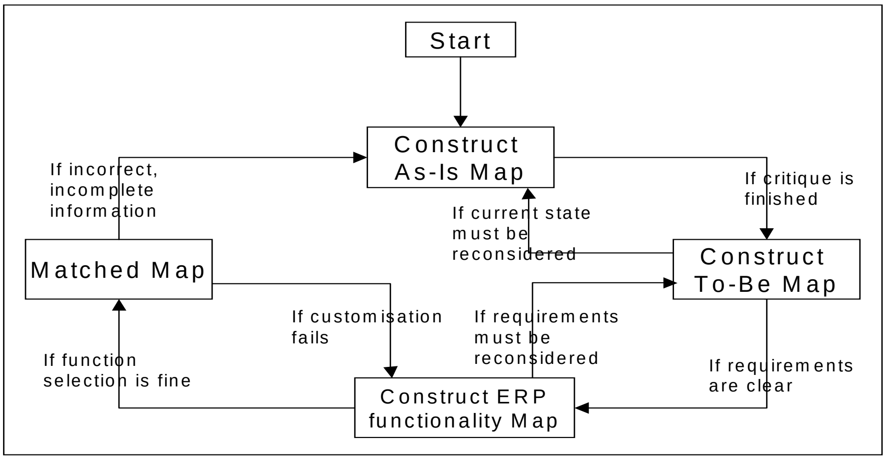

Figure 4 The customising process

The *As-Is map* resulting from (a) abstracts current practice from the organisation to describe the currently achieved goals. It serves as a support for critiquing the current situation and thereby identifying customer requirements for the *To-Be map*. Additionally, it also serves as a reference to evaluate the new solution against current practice. The *To-Be map* reflects the customer requirements that the organisation would like to satisfy by acquiring and customising a COTS product. These requirements are expressed in terms of goals and their strategies that are to be supported by the COTS product. The *COTS map* specifies the functionalities of the COTS product in terms of intentions and associated strategies that the product supports. It might be seen as the set of business requirements that the COTS product is able to satisfy.

The heart of the customising process is to construct the *Matched map* which is the output of the customising process and the input of the installation process. Most of intentions and strategies of the *Matched Map* are obtained from the COTS map and match the *To-Be* requirements. Others may not be available in the COTS map and will require in-house development. In such a case, the *Matched map* helps in their identification. On the contrary, all the intentions and strategies of the COTS map may not be included in the *Matched map*. This corresponds to the COTS functionality that is not matching the requirements in the *To-Be map*. In order to construct the *Matched map* we defined two operators: *prune* and *graft* which respectively eliminates and introduces elements (intentions, strategies or sections) of the map. Pruning eliminates those parts of the map that are irrelevant whereas grafting introduces elements to satisfy specific needs that are not available in the original map.

Let us illustrate the process with the *Energy* COTS product. The As-Is and To-Be maps are shown in Fig.5 &6 respectively.

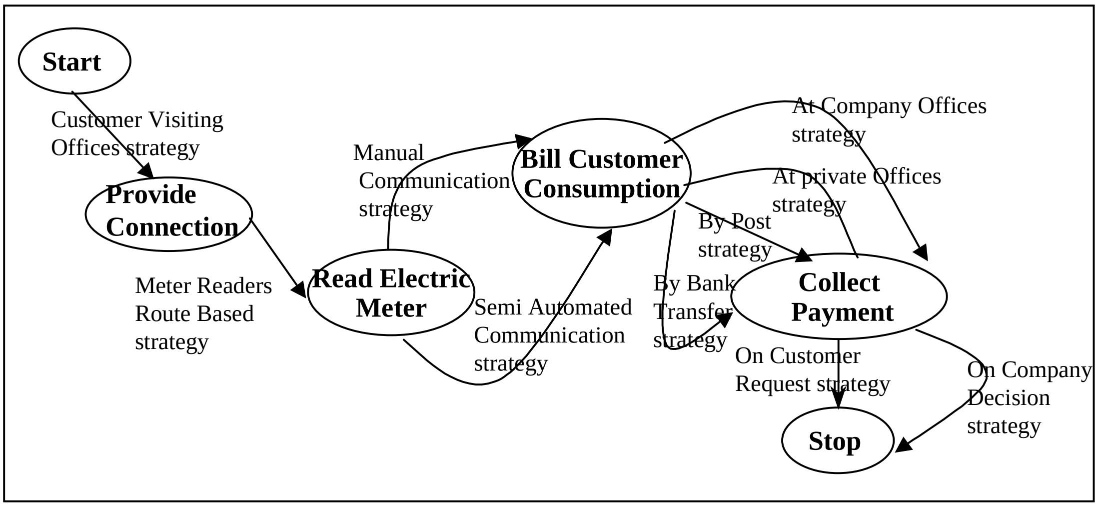

Figure 5 The As-Is map

Fig.5 shows the constructed As-Is map which represents the way in which customers are currently serviced. This is achieved by a sequence of four intentions. In order to "*Provide Connection*" customers visit the companies offices. This leads to long customer queues. It is also found that providing a connection itself takes too long. Once a connection is provided, "*Billing Customer Consumption*" of electricity is based on "*Reading Electric Meter*". Meter readers according to a pre-defined plan do the latter. It is found that there are too many errors in the bill and bills are inordinately delayed. In order to "*Collect Payment*" the company offers to its customers a number of strategies. Three of these are according to the conventional cash payment mode. Payments made to Post offices and Private offices are too expensive and not efficient. Finally, electricity supply can be "*Stopped*" either on customer request or by company's decision.

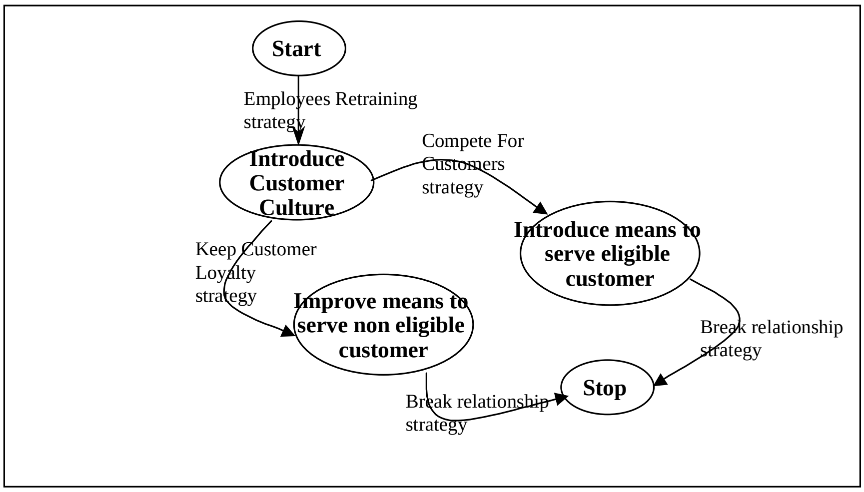

Figure 6 The To-Be map

Under the anticipated deregulation, the company needs to become more customer-friendly. This is reflected in the To-Be map shown in Fig. 6 by the intention "*Introduce Customer Oriented Culture*" achieved through the "*Retraining Employees strategy*". The deregulation introduces two kinds of customers, those who have the choice of their electricity supplier (called eligible customers) and those who do not (the non-eligible customers). The company must re-orient its processes to win over as many eligible customers as possible. This is reflected in the To-Be map by the intention "*Introduce Means To Serve Eligible Customer*" under the "*Compete For Customer strategy*". To protect its customer base for the future the company has to serve its non-eligible customers better. This leads to the intention "*Improve Means to Serve Non-eligible Customers*" under the "*Keep Customer Loyalty strategy*".

The *Energy* COTS product introduced in Fig. 2 was selected as a good candidate to meet the requirements. This selection was based on its description in Table 1. The matched map resulting from the *Match Maps* process is shown in Figure 7. It is obtained by a customization and integration of the To-Be and *Energy* maps.

Starting from the To-Be map, the *Energy* map is examined and it is found that the two sections from *Start* to "*Serve Customer Request*" are driven by strategies, which are very close to the ones in the To-Be map for serving eligible and non-eligible customers. This leads to the acceptance of "*Serve Customer Request*" and its associated strategies. A detailed examination of the strategies now leads to the rejection of the "*Captive strategy*" and also shows that the "*Competitive strategy*" is a bundle of "*Marketing strategy*", "*Change Culture strategy*", "*Contracting strategy*", and "*Intelligent Front Desk (IFD) strategy*".

The "*Change Culture strategy*" to "*Serve Customer Request*" can obviously be related to the intention "Introduce Customer Oriented Culture" in the To-Be map. This confirms the need of the section <*Start, Serve Customer Request, Change Culture strategy*> in Figure 7. The "*Marketing strategy*" and the "*Contracting strategy*" to "*Serve Customer Request*" are not explicitly expressed in the To-Be map. Rather they are suggested by the COTS map. Evidently "*Introduce Customer Oriented Culture*" requires both marketing and new forms of contracting. In the former case the company is marketed to potential customers whereas in the latter contracts are developed to match the requirements of specific customer-profiles and are offered to them. This leads to the sections <*Start, Serve Customer Request, Marketing strategy*> and<*Start, Serve Customer Request, Contracting strategy*> in Figure 7. The adoption of the IFD strategy suggests special treatment for some important customers leading to "*IFD at Customer Premises strategy*" complementary to "*IFD at Company Premises strategy*". Then, the two sections <*Start, Serve Customer Request, IFD at Company Premises strategy* > and <*Start, Serve Customer Request, IFD at Customer Premises strategy* > are included in Figure 7. Whereas the other strategies promote customer-orientation, the IFD strategies directly contribute to reducing connection delays and customer queues at offices. These were some of the difficulties of the current practice discussed earlier. The rejection of "*Captive strategy*" is motivated by the anticipation that non-eligible customers will not stay captive very long. Consequently, it is best to treat them in the same way as eligible customers.

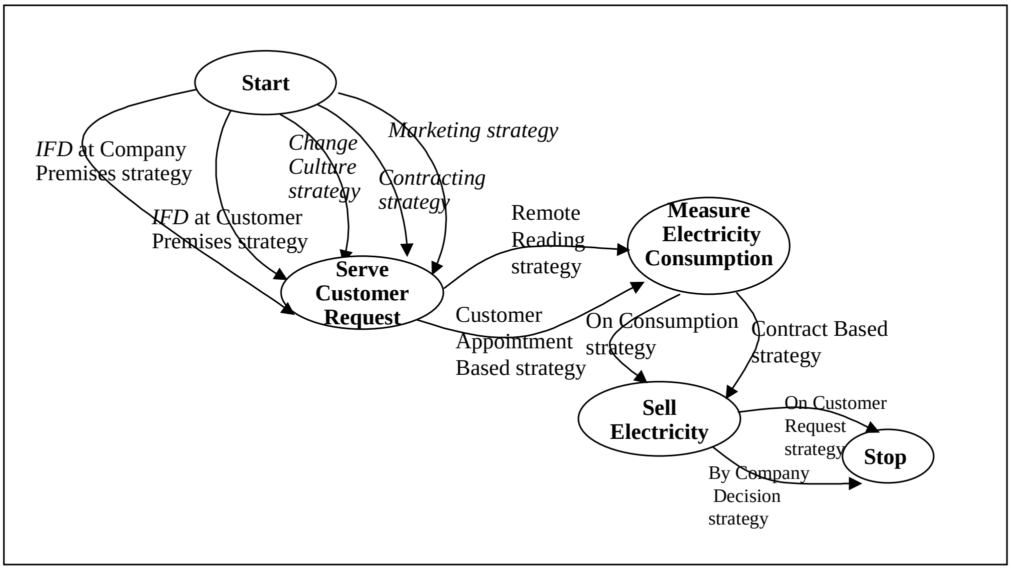

Figure 7 The matched map

The COTS product contains the "*Sell Electricity*" intention as well. An examination of this intention shows that it is dealing with electricity consumption, billing and payments. These are integral parts of customer servicing as can be confirmed from the As-Is map. In order for the To-Be map to be compliant with the As-Is map, it is clear that the intention "*Sell Electricity*" must be retained in the solution. The COTS offers two strategies for this intention. The "*Advance Payment strategy*" is found to be difficult to implement due to customer resistance to this change and high cost of replacing old meters with new ones. Therefore, the "*Credit strategy*" is accepted.

The matching process proceeds with the examination of the refinement of the section <*Serve Customer Request, Sell Electricity, Credit strategy*> (Fig. 3). Fig. 7 contains the two intentions "*Measure Electricity Consumption*" and "*Sell Electricity*" found in Figure 3 as well as the multi-thread between these two. This leads to the two sections <*Measure Electricity Consumption, Sell Electricity, Contract Based strategy*> and <*Measure Electricity Consumption, Sell Electricity, On consumption strategy*>. Out of the various strategies for "*Measure Electricity Consumption*" in Fig. 3, only *the "Remote Reading strategy*" is selected. The strategy is innovative and implies a large investment, as all current electric meters have to be replaced by new ones. However the company believes that this investment is worse doing as it will give her the possibility to sell new services such as remote heating or cooling to its customers; therefore providing a competitive advantage to its competitors. Besides, remote meter reading will avoid the mistakes in billing as well as the delay in billing which are critiques emerging from the current practice. On examination, it was found that remote reading would not work in all cases and a completely new *strategy "Customer Appointment Based strategy*" is introduced. This leads to the two sections <*Serve Customer Request, Measure Electricity Consumption, Remote Reading strategy*>, <*Serve Customer Request, Measure Electricity Consumption*, *Customer Appointment Based strategy* > in Fig. 7. Lastly, it can be noticed that in Figure 7 the strategies to *Stop* are those that are contained in the ESM map since these are found to be compliant with the As-Is map.

The foregoing example brings out that:

- The As-Is is useful since it provides domain knowledge, which helps in understanding differences, finding incompleteness, reusing solutions and evaluating the future solution against critiques of the current practice.
- New strategies can be introduced in the matched map directly. These are not supported by any component and have to be developed in-house. Though not brought out by the example, it is possible for this to happen for intentions as well.
- Initial requirements in the To-Be map are refined and new ones discovered while examining functionality offered by components.
- The matching process might end up suggesting several alternative matched maps. This might occur when the decision for component selection and/or assembly requires further evaluation such as cost/benefit analysis, risk analysis etc.

7. Conclusion

Starting with the observation that developing purposeful systems is a current, unsolved problem, we argued in favour of goal-oriented requirements engineering and introduced the *Map* representation system as a means to (a) complete goals with strategies to achieve them and (b) to express goal dependencies. Our belief is that these two extensions are useful to reason about the alignment between organisational goals and system functionality.

We showed that by expressing system functionality in goal-strategy terms, the map provides a representation of the functionality in a language that is easily understood by an organisation. This helps in expressing the relationship between the organisational goals and system functionality, the *fitness relationship,* in a straight way which eases resolving the alignment problem. We showed how this expression helps in the case of COTS installation. The map representation system nudges an organisation to looking at its systems in a holistic way rather than in narrow operational terms. We illustrated how this helps in customising the COTS product offer but in high level goal-strategy rather than in low level functionality terms.

The map provides a basis for a two-way interchange between a goal model and a system functionality model; for example from the COTS functionality to organisational requirements and vice-versa. This is facilitated by the level at which the interchange takes place, organisational goals-strategies and COTS goals-strategies. As a result, the map has the potential to better align organisational needs with COTS offerings.

Finally, it is clear that the map needs to be supported by a mechanism that provides guidance (a) to construct a map and check its correctness and (b) to refine maps, (c) to support the alignment process in different situations such as organisational change, ERP installation, COTS product adaptation, component or services composition to match organization needs. This will form the topic of future work. A future step is also, in a longer term to abstract from experience a collection of alignment patterns that could be made publicly available.

Acknowledgments: The author wants to express her deep and kind thanks to Francois Bodart for having inspired so many ideas of her in the field of Information System engineering. She hopes that he might accept this ‘to-be-developed’ concept of fitness relationship as a gift for his retirement.

References

[Anton98] A.I. Anton C. Potts (1998). *The Use of Goals to Surface Requirements for Evolving Systems*. In Proceedings of IEEE International Conference on Software Engineering, IEEE Computer Society.

[Arsanjani01] A. Arsanjani, J. Alpigini (2001). *Using Grammar-oriented Object Design to Seamlessly Map Business Models to Component-based Software Architectures.* In Proceedings of the International Symposium of Modelling and Simulation, May 16-18, 2001, Pittsburgh, PA, USA, pp 186-191.

[Eatock00] J. Eatock, G. M. Gliaglis R. J. Paul, A. Serrano (2000). *The Implications of Information Technology Infrastructure Capabilities for Business Process Change Success*. Henderson, P. (Ed.) Systems Engineering for Business Process Change. 2000. London: Springer-Verlag.

[ESI96], European Software Institute (1996).*European User survey analysis*, Report USV_EUR 2.1, ESPRIT Project.

[Gotel&94] Gotel, O., and Finkelstein A. (1994). *Modelling the contribution structure underlying requirements*, in Proceedings of the First Int. Workshop on Requirements Engineering : Foundation of Software Quality, Utrech, Netherlands.

[Jarke&93] Jarke, M., and Pohl, K., (1993). Establishing visions in context: towards a model of requirements processes. Proceedings of the 12th Intl. Conf. Information Systems, Orlando, USA.

[Lamsweerde00] Lamsweerde, A. v., (2000). *Requirements engineering in the year 00: A research perspective*. In Proceedings 22nd International Conference on Software Engineering, Invited Paper, ACM Press.

[Lamsweerde&00a] Lamsweerde, A. v., and Letier, E., (2000). *Handling obstacles in goal-oriented requirements engineering*, IEEE Transactions on Software Engineering, Special Issue on Exception Handling, Vol. 26 No. 10, October 2000, pp. 978-1005.

[Letier02] E. Letier, A. van Lamsweerde (2002). *Agent-Based Tactics for Goal-Oriented Requirements Elaboration.* In Proceedings of ICSE'02 - 24th International Conference on Software Engineering, ACM Press.

[Loucopoulos94] Loucopoulos, P, (1994). *The f^3 (from fuzzy to formal) view on requirements engineering*. Ingénierie des systèmes d'information, Vol. 2 N° 6, pp. 639-655.

[Paech02] B. Paech, A.H. Dutoit, D. Kerkow, A. von Knethen (2002). *Functional Requirements, non-Functional Requirements, and Architecture Should not Be Separated*. In Proceedings. of REFSQ'02. International Workshop on Requirements Engineering : Foundation for Software Quality. Essen, Germany, Sept 2002.

[Pohl 96] Pohl K., (1996). *Process centred requirements engineering*, J. Wiley and Sons Ltd.

[Potts97] C. Potts, I. (1997). Abstraction and Context in Requirements Engineering: A Synthesis of Goal Refinement and Ethnography. Annals of Software Engineering, Vol. 3 pp. 23-61.

[Ramesh95] Ramesh, B., Powers, T., Stubbs, C., and Edwards, M., (1995). *Implementing requirements traceability : a case study*, in Proceedings of the 2nd Symposium on Requirements Engineering (RE'95), pp89-95, UK.

[Reich96] B. H. Reich, I. Benbasat, (1996). *Measuring the Linkage Between Business and Information Technology Objectives.* MIS Quarterly, 1996, May, pp. 55-81.

[Robinson89] Robinson, W. N., (1989). *Integrating multiple specifications using domain goals*, Proc. IWSSD-5 - 5th Intl. Workshop on Software Specification and Design, IEEE, 219-225.

[Rolland&98] C. Rolland, C. Souveyet, C. Ben Achour (1998). *Guiding Goal Modeling Using Scenarios*. IEEE Transactions on Software Engineering, Special Issue on Scenario Management, Vol. 24, No 12, p. 1055-1071. December 1998.

[Rolland&99] Rolland, C., Grosz, G., and Kla, R., (1999) *Experience with goal-scenario coupling. in requirements engineering*, Proceedings of the Fourth IEEE International Symposium on Requirements Engineering, Limerik, Ireland.

[Rolland&00] C. Rolland, N Prakash, (2000). *Bridging the Gap Between Organisational Needs and ERP functionality.* Requirements Engineering Journal, Springer Verlag. 5:180.193. 2000.

[Rolland&01] C. Rolland, N. Prakash. *Matching ERP System Functionality to Customer Requirements.* In Proceedings of the 5th IEEE International Symposium on Requirements Engineering, Toronto, Canada. August 27-31, 2001

[Rolland&03] C. Rolland, C. Salinesi, A. Etien (2003). *Eliciting Gaps in Requirements Change*, Requirements Engineering Journal, Volume 8, Number 2.

[Standish95] Standish Group, (1995). *Chaos*. Standish Group Internal Report, http://www.standishgroup.com/chaos.html

[Yu94] Yu, E., (1994). *Modelling strategic relationships for process reengineering*. Ph.D. Thesis, Dept. Computer Science, University of Toronto, Dec. 1994.

[Yu01] E. Yu, (2001°. *Agent Orientation as a Modelling Paradigm*. Wirtschaftsinformatik. 43(2) April 2001. pp. 123-132.